Learn new features of Red Hat Enterprise Linux 7.1 on IBM z Systems **by Examples**
Session# 17493

Filipe Miranda <fmiranda@redhat.com>
Global Lead for Red Hat Products on IBM z Systems and Power Systems
Red Hat Inc.

Session topics

Red Hat in a Nutshell
- Open Source Development Model
- Red Hat Subscription Model
- JBoss EAP and BRMS for RHEL on z Systems

RHEL 7.1 z Systems Features Summary

IBM Redbooks Publications

Virtualization Cookbook for IBM z Systems Volume 2:
Red Hat Enterprise Linux 7.1
- Installation process review
  - Manual
  - Kickstart
- Automated LUN scanning for FCP (NPIV)
- systemd
  - Journal

Loading different system targets
Rescue RHEL - z/VM

Red Hat in a Nutshell

The FIRST

$1.8 BILLION DOLLAR

OPEN SOURCE COMPANY in the WORLD

Source: Red Hat Inc.

MORE THAN

90%

of the

FORTUNE 500

use

RED HAT

PRODUCTS & SOLUTIONS. *

75+

WORLDWIDE OFFICES

35

COUNTRIES

S&P

500

COMPANY

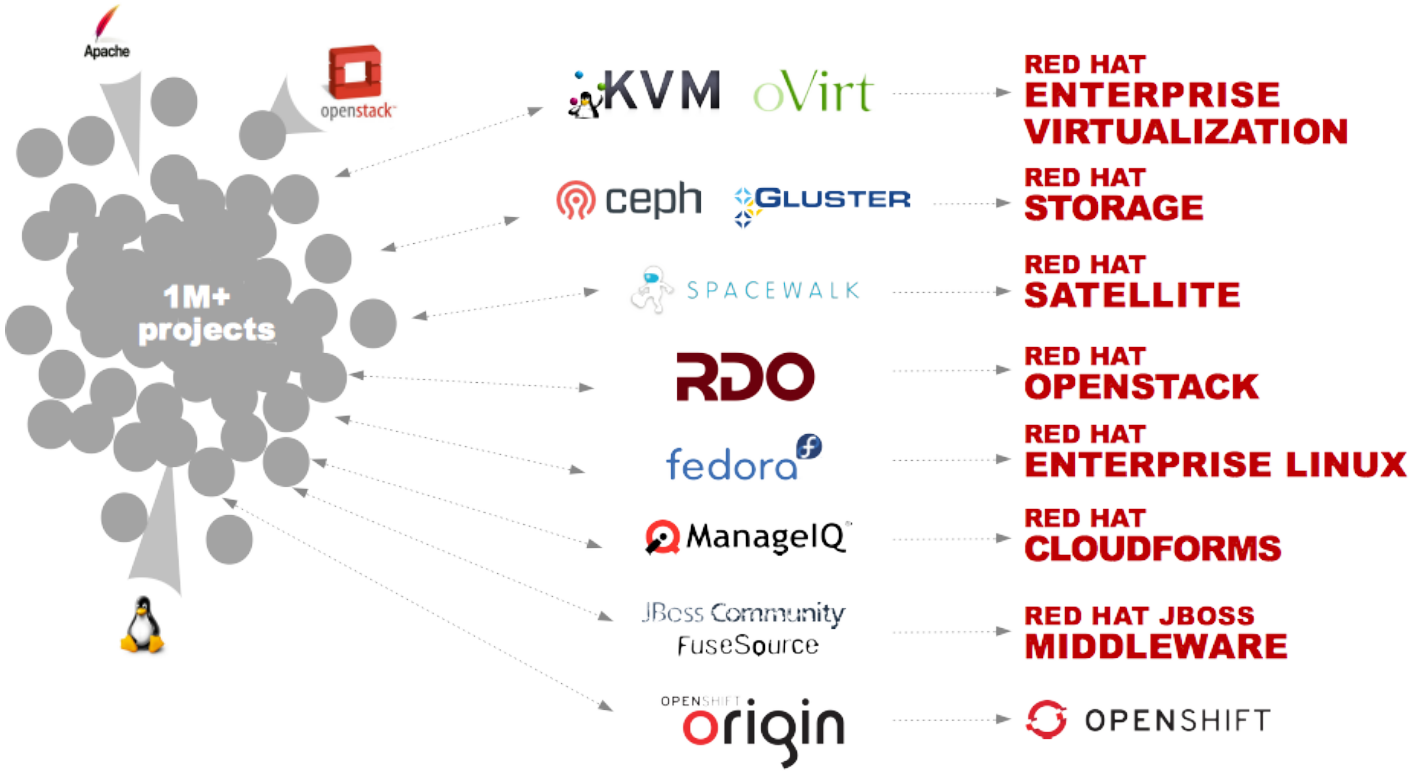
NYSE:

RHT

AWARD-WINNING

SOLUTIONS

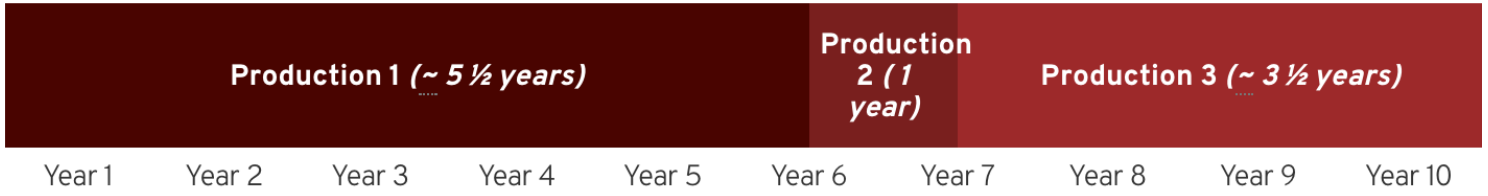
Red Hat Open Source Development Model



* www.blackducksoftware.com/oss-logistics/choose

Standard Subscription Production Support

- The Red Hat Enterprise Linux 5, 6, and 7 Life Cycle*:



- Lifecycle Dates:

Version	General Availability	End of Production 1	End of Production 2	End of Production 3 (End of Production Phase)
5	March 15, 2007	January 8, 2013	January 31, 2014	March 31, 2017
6	November 10, 2010	Q2 of 2016	Q2 of 2017	November 30, 2020
7	June 10, 2014	Q4 of 2019	Q4 of 2020	June 30, 2024

JBoss for RHEL on z

Red Hat Enterprise Linux optimized to z Systems (I)

JBoss EAP and BRMS adding Value to your solution (II)

Take advantage of the IBM JDK on Red Hat Enterprise Linux for IBM z Systems

(I) http://www-03.ibm.com/systems/z/os/linux/resources/testedplatforms.html

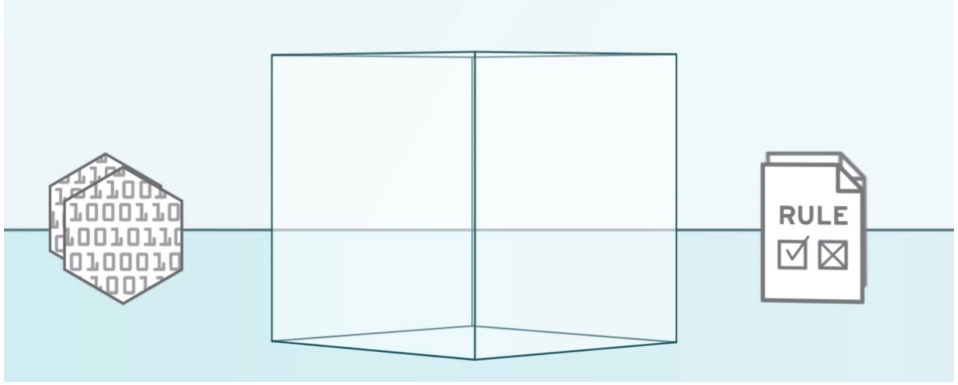
(II) https://access.redhat.com/site/articles/111663

JBoss BRMS for RHEL on z

What is Red Hat JBoss BRMS?

Red Hat® JBoss® BRMS is a comprehensive platform for business rules management, business resource optimization, and complex event processing (CEP).

- Deploy decision services across physical, virtual, and cloud environments.
- Improve business agility.
- Make consistent and efficient decisions.
- Quickly build resource optimization solutions.
- Shorten development cycles for faster time to market.



https://www.redhat.com/en/technologies/jboss-middleware/business-rules

RHEL 7.1 s390x Features

zFCP Specific

- End-To-End data consistency checking for zfcp - Production Ready -
- Exploitation of Data Routing for FCP
- Automated LUN scanning for NPIV only

Memory Specific

- Support of transparent large pages for **z Systems**
- libhugetlbfs support for **z Systems**
- Cross Memory Attach for **z Systems**
- Transactional memory support (for zEC12 and newer)
- Implement write protection based dirty page detection

Network Specific

- Enhancement in the configuration tool for **z Systems** network devices
- IPv6 support for qetharp tool
- Support of VEPA (Virtual Ethernet Port Aggregator)

RHEL 7.1 s390x Features

DASD Specific

- Safe offline interface for DASD devices
- Enhanced DASD statistics for PAV and HPF
- DASD sanity check to detect path connection error
- Improve performance of dasdfmt - Production Ready -

Crypto Specific

- Support for Crypto Express4S
- Crypto adapter resiliency
- Tolerance for Crypto Express5S - Production Ready -

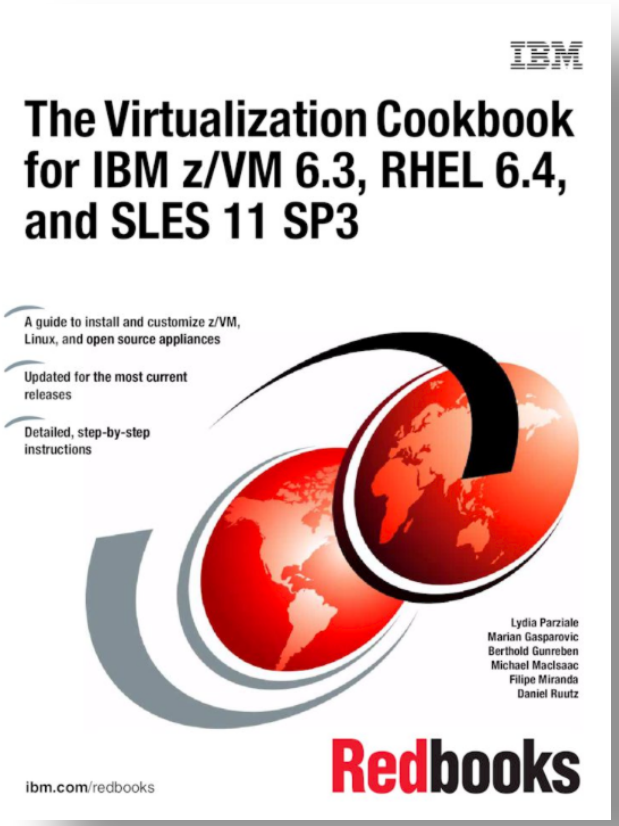
RHEL 7.1 s390x Features

All other features

- zipl to automatically calculate boot device ramdisk address
- Optimized compression library zlib for Linux on z Systems
- Kernel support to improve Java performance for Linux on z Systems - Production Ready -
- Enable LLVM pipe for z Systems
- Architecture level set for IBM System z196 and newer
- Support for zEC12 Flash Express - Production Ready -
- Provide PCHID mapping
- Fuzzy live dump for z Systems - Production Ready -
- Two Stage Dumper - Production Ready -
- Linux support for concurrent Flash MCL updates - Production Ready -

IBM Redbooks Publications

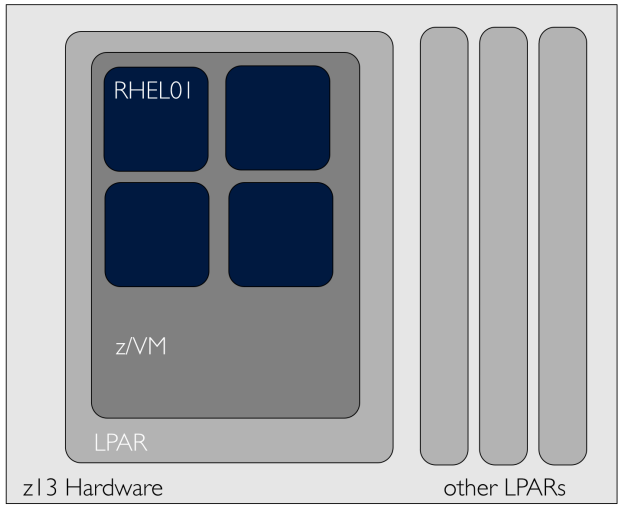
2013



2015

http://www.redbooks.ibm.com

Installation Overview



Linux Desktop

RHEL 7.1 manual installation process

These files can be found at the images directory from the RHEL7.x installation tree

```
kernel.img
initrd.img
redhat.exec
generic.prm
```

Highly recommended to edit the PRM file locally in your desktop and then copy it to the z/VM Virtual Machine.

This is how the stock generic.prm file looks like:

```
ro ramdisk_size=40000 cio_ignore=all,!condev
```

RHEL 7.1 FCP, ECKD/EDEV DASD

Examples of GENERIC.PRM:

- FCP:

ro ramdisk_size=40000 cio_ignore=all,!condev
ip=9.12.7.99::9.12.4.1:20:vmlnx2-4.itso.ibm.com:enccw0.0.0600:none
rd.znet=qeth,0.0.0600,0.0.0601,0.0.0602,layer2=1
nameserver=9.12.6.7 nameserver=9.12.6.6
rd.zfcp=0.0.fc00,0x500507630500c74c,0x4010401800000000
rd.zfcp=0.0.fd00,0x5005076305 10c74c,0x4010401800000000
inst.repo=ftp://9.12.7.96/pub/rhel71 ks=ftp://9.12.7.96/pub/kickstart/linux2-ks.cfg
inst.cmdline

- ECKD DASD/EDEV:

ro ramdisk_size=40000 cio_ignore=all,!condev
ip=9.12.7.98::9.12.4.1:20:vmlnx2-3.itso.ibm.com:enccw0.0.0600:none
rd.znet=qeth,0.0.0600,0.0.0601,0.0.0602,layer2=1
nameserver=9.12.6.7 nameserver=9.12.6.6
rd.dasd=0.0.0100
inst.repo=ftp://9.12.7.96/pub/rhel71 ks=ftp://9.12.7.96/pub/kickstart/linux1-ks.cfg
inst.cmdline

Copying files to a z/VM guest

To copy files over to z/VM, open a Linux terminal on your desktop:

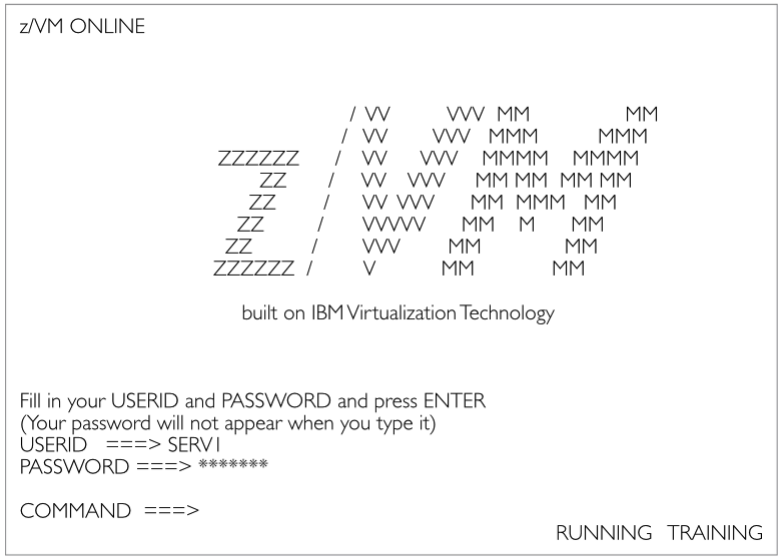
```
# ftp <FQDN for the z/VM>
    login <login>
    password <password>
```

Example:

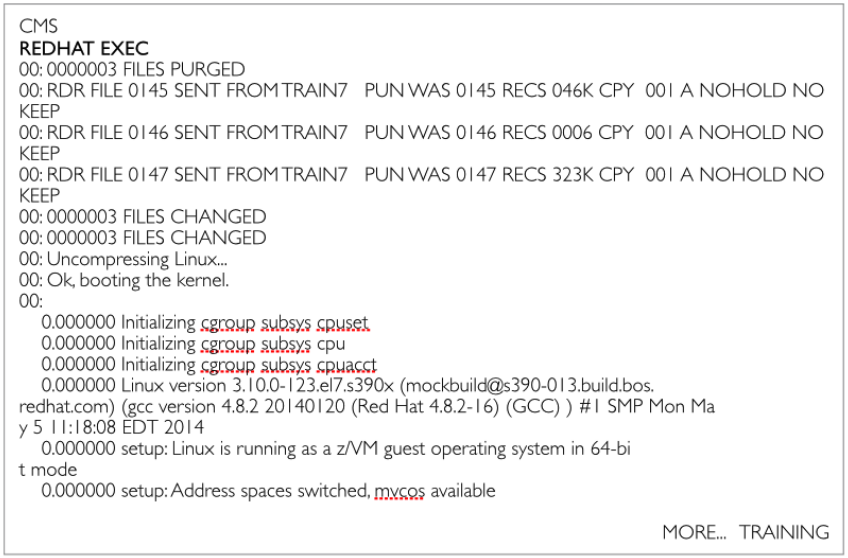
```
site fix 80
bin
put kernel.img KERNEL.IMG
put initrd.img INITRD.IMG
ascii
put redhat.exec REDHAT.EXEC
put generic.prm GENERIC.PRM
```

RHEL 7.1 installation process

1



2



RHEL 7.1 installation process

3

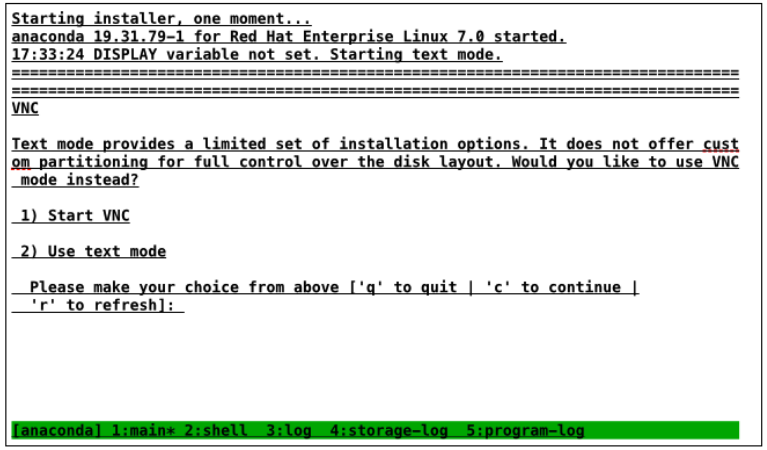
```
Starting sshd to allow login over the network.

anaconda: Starting installer, one moment...
anaconda: 17:33:22 Please ssh install@serv1.redhat.com (10.16.10.71) to begin the install.

_

                                                    RUNNING  TRAINING
```

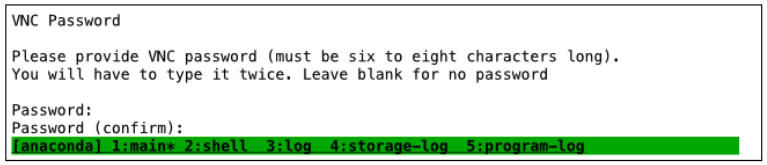
4

```
~ fmiranda$ ssh install@serv1.redhat.com
```

RHEL 7.1 installation process

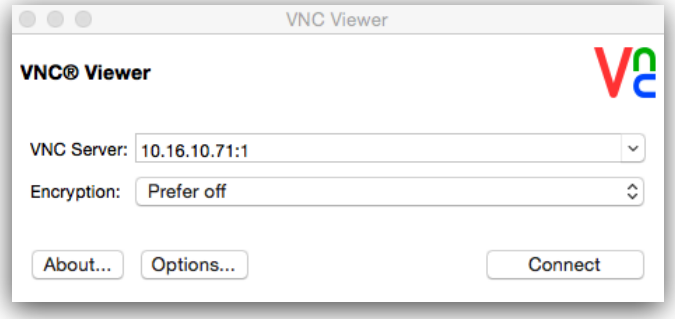
5



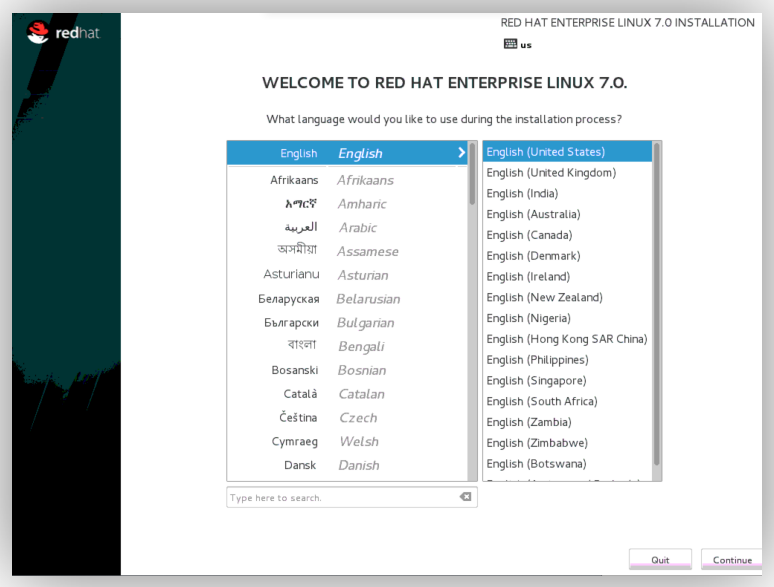


RHEL 7.1 installation process

6



7



Kickstart format

```
#version=RHEL7
 # System authorization information
 auth --enableshadow --passalgo=sha512
 # RHEL7 TEMPLATE KICKSTART FOR DASD
 # Use network installation
 install
 url --url="ftp://9.12.7.96/pub/rhel71"
 # Use text mode install
 text
 ignoredisk --only-use=dasda
 # Keyboard layouts
 keyboard --vckeymap=us --xlayouts='us'
 # System language
 lang en_US.UTF-8
 # Network information
network --bootproto=static --device=enccw0.0.0600 --gateway=9.12.4.1 --ip=9.12.7.98 --nameserver=9.12.6.6,9.12.6.7 --netmask=255.255.240.0 --noipv6
--activate --hostname=linux1.itso.ibm.com
# Root password
rootpw --iscrypted $6$pr46QGx7PLwzthjk$41E7GLPSsD//jHPwbQc7/CAG2SSQSkGg/pcveQUXz2IIVL0LCXH2So8n.e1 rFMjqLrfMYWifE7qY2NFfygedw/
# System timezone
timezone America/New_York
# Skip X
skipx
# System bootloader configuration
bootloader --location=mbr --append="hvc_iucv=8 console=hvc0 console=ttyS0" zerombr
# Partition clearing information
clearpart --all
autopart --type=lvm
reboot
```

Kickstart Package and Post Scripting Sections

- Package selection

```
%packages
  @core
  kexec-tools
  %end
```

- Post Scripting

```
%post --log=/root/post.log
  # Enable the DVD repo
  cat > /etc/yum.repos.d/dvd.repo <<EOF
  [DVD]
  name= RHEL7.1 DVD ISO
  baseurl=ftp://9.12.7.96/pub/rhel71/
  enable=1
  gpgcheck=1
  EOF
```

Kickstart Advanced Post Scripting

Post Scripting

```
#import RedHat GPG key to verify packages authenticity during yum package install
rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release

# Enable the VDISKs for swap
echo 'persistent_policy=by-path' >> /etc/dracut.conf
dracut -f
zipl
cio_ignore -r 0.0.0300
cio_ignore -r 0.0.0301
chccwdev -e 0.0.0300
chccwdev -e 0.0.0301
echo '0.0.0300' >> /etc/dasd.conf
echo '0.0.0301' >> /etc/dasd.conf
echo '/dev/disk/by-path/ccw-0.0.0300-part1 swap swap pri=5 0 0' >> /etc/fstab echo '/dev/disk/by-path/ccw-0.0.0301-part1 swap swap pri=4 0 0' >> /
etc/fstab

#Enable IUCV hvc0 for the Linux system
ln -s /etc/systemd/system/serial-getty\@hvc0.service /lib/systemd/system/serial-getty\@.service

%end
%addon com_redhat_kdump --enable --reserve-mb='4096' %end
```

Kickstart with FCP devices (optional)

```
...
# Use text mode install
text
zfcp --devnum=fc00 --wwpn=500507630500c74c --fcplun=0x4010401800000000
zfcp --devnum=fd00 --wwpn=50050763051 0c74c --fcplun=0x4010401800000000
# Keyboard layouts
keyboard --vckeymap=us --xlayouts='us'
# System language
...
```

Automated LUN Scanning (NPIV)

Example on how simple it is to work with FCP devices:

1 .Unblock the devices

```
# cio_ignore -r fc00
# cio_ignore -r fd00
```

2 .Enable the devices

```
# chccwdev -e fc00
# chccwdev -e fd00
```

3. LUN already detected on all paths

```
# lsluns
  Scanning for LUNs on adapter 0.0.fc00
        at port 0x500507630500c74c:
              0x4010401700000000
        at port 0x50050763050bc74c:
              0x4010401700000000
  Scanning for LUNs on adapter 0.0.fd00
        at port 0x5005076305 10c74c:
              0x4010401700000000
        at port 0x5005076305 1bc74c:
              0x4010401700000000
```

Automated LUN Scanning (NPIV)

4. Enable multipath:

```
# cp  /usr/share/doc/device-mapper-multipath-0.4.9/multipath.conf   /etc/multipath.conf

# systemctl start multipathd
# systemctl enable multipathd

# multipath -ll
                    mpatha (36005076305ffc74c0000000000001017) dm-2 IBM
                    size=10G features='1 queue_if_no_path' hwhandler='0' wp=rw
                    `-+- policy='service-time 0' prio=1 status=active
                      |- 0:0:0:1075265552 sda 8:0  active ready running
                      |- 0:0:1:1075265552 sdb 8:16 active ready running
                      |- 1:0:0:1075265552 sdc 8:32 active ready running
                      `- 1:0:1:1075265552 sdd 8:48 active ready running
```

5. Make the FCP devices persistent:

```
# lszfcp -D | awk '{ print $1 }' | sed -e 's/\// /g' >> /etc/zfcp.conf
```

6. Partition the multipath device:

```
# parted -s /dev/mapper/mpatha mklabel msdos mkpart primary 0% 100%
```

systemd overview

systemd is a system and services manager that replaced upstart in RHEL. systemd is the first user space process the kernel starts when booting.
This process is responsible for starting all the services and their dependencies that allow the system to act as a server. systemd uses units that can be dependent on other units.

There are different unit types such as:

- Service units, used to start services
- Socket units, which allow socket based activations
- Device units, which trigger reactions for devices as they appear or disappear
- Mount point units, that control mount points
- Target units, which allow grouping of units to act as synchronization points

systemd script example

root@localhost:~/systemd | root@localh

```
[root@localhost ~]# cat /usr/lib/systemd/system/sshd.service
[Unit]
Description=OpenSSH server daemon
After=syslog.target network.target auditd.service

[Service]
EnvironmentFile=/etc/sysconfig/sshd
ExecStartPre=/usr/sbin/sshd-keygen
ExecStart=/usr/sbin/sshd -D $OPTIONS
ExecReload=/bin/kill -HUP $MAINPID
KillMode=process
Restart=on-failure
RestartSec=42s

[Install]
WantedBy=multi-user.target
[root@localhost ~]#
```

systemd example versus system V

```
File  Edit  View  Search  Terminal  Help
[root@samba4 ~]# cat /etc/redhat-release
Red Hat Enterprise Linux Server release 6.4 (Santiago)
[root@samba4 ~]# cat /etc/init.d/sshd  | wc
     234     666    4534
```

```
root@localhost:~/systemd                    root@localhost:~
[root@localhost ~]# cat /etc/redhat-release
Red Hat Enterprise Linux Server release 7.0 (Maipo)
[root@localhost ~]# wc /usr/lib/systemd/system/sshd.service
 15  21 334 /usr/lib/systemd/system/sshd.service
```

Managing services with systemd

List active service units (running services):

```
# systemctl -t service
UNIT                    LOAD   ACTIVE SUB     DESCRIPTION
after-local.service     loaded active exited  /etc/init.d/after.local Compatibility
cpi.service             loaded active exited  LSB: Set Control Program
cron.service            loaded active running Command Scheduler
dbus.service            loaded active running D-Bus System Message Bus
...
```

List failed service units:

```
# systemctl -t service --state=failed
```

systemd commands

Query the status of a service unit:

```
# systemctl status sshd.service
sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled)
   Active: active (running) since Thu 2015-04-23 16:40:03 EDT; 24h ago
 Main PID: 1454 (sshd)
   CGroup: /system.slice/sshd.service
           ··1454 /usr/sbin/sshd -D

Apr 23 16:40:03 linux2.itso.ibm.com systemd[1]: Started OpenSSH server daemon.
Apr 23 16:40:03 linux2.itso.ibm.com sshd[1454]: Server listening on 0.0.0.0 ...
Apr 23 16:40:03 linux2.itso.ibm.com sshd[1454]: Server listening on :: port 22.
Apr 23 16:58:13 linux2.itso.ibm.com sshd[2906]: Accepted password for ken f...2
...
```

systemd commands

Stop, start, and restart a service unit:

```
# systemctl stop vsftpd.service
# systemctl start vsftpd.service
# systemctl restart vsftpd.service
```

You can also omit the suffix and specify multiple units:

```
# systemctl restart vsftpd sshd
```

Reload a service unit

```
# systemctl reload sshd
```

Not all services support that

systemd commands

List installed service units:

```
# systemctl list-unit-files -t service
```

List enabled service units:

```
# systemctl list-unit-files -t service --state=enabled
UNIT FILE                                   STATE
btrfsmaintenance-refresh.service            enabled
cio_ignore.service                          enabled
cron.service                                enabled
dm-event.service                            enabled
getty@.service                              enabled
```

Disable a service:

```
# systemctl disable vsftpd
```

Query the default target the system boots into:

```
# systemctl get-default
```

Switch to a target:

```
# systemctl isolate multi-user.target
```

systemd/SysV table

systemd Target	SysV Runlevel	Notes
poweroff.target	0	Halts the system
rescue.target	1, s, single	Single user mode that provides a base system and a rescue shell
multi-user.target	2,3,4	Multi-user, non-graphical but with Network and Services running
graphical.target	5	Multi-user, Graphical
reboot.target	6	Reboot the system
emergency.target	emergency	Emergency shell. This is a special systemd target unit that can be be specified as a kernel command line argument: systemd.unit=emergency.target

systemd Journal

The journal is part of systemd and provides a modern logging mechanism. It allows to capture Kernel log messages, regular syslog messages, the stdout/stderr written by services, and messages from the early boot stages.

Viewing the journal

```
# journalctl
-- Logs begin at Wed 2015-04-22 14:59:04 EDT, end at Wed 2015-04-22 16:44:17 EDT.
Apr 22 16:43:51 linux1.itso.ibm.com systemd-journal[64]: Runtime journal is using
Apr 22 16:43:51 linux1.itso.ibm.com systemd-journal[64]: Runtime journal is using
Apr 22 16:43:51 linux1.itso.ibm.com kernel: Initializing cgroup subsys cpuset
...
```

Filtering Journal

Filtering the journal

Show the log messages of the current boot (filters out the messages from previous boots):

 # journalctl -b

Show today's log messages:

 # journalctl --since today

Show kernel messages of the current boot only:

 # journalctl -b -k

Only show errors:

 # journalctl -p err

Show the log messages of a specific unit (like the sshd.service unit):

 # journalctl -u sshd.service

Load system targets on RHEL z/VM - FCP

IPL your Linux server from the 3270 console using the following commands:

```
===> cp set loaddev portname 50050763 0500C74C lun 40104018 00000000 scpdata
systemd.unit=rescue.target
===>ipl fc00

00: HCPLDI2816I Acquiring the machine loader from the processor controller.
00: HCPLDI2817I Load completed from the processor controller.
00: HCPLDI2817I Now starting the machine loader.
....
es: 258048
[    0.000000] Kernel command line: rd.znet=qeth,0.0.0600,0.0.0601,0.0.0602,laye
r2=1 rd.lvm.lv=rhel_linux2/root cio_ignore=all,!condev console=hvc0 root=/dev/ma
pper/rhel_linux2-root console=ttyS0 rd.lvm.lv=rhel_linux2/swap rd.zfcp=0.0.fd00
0x500507630510c74c,0x4010401800000000 rd.zfcp=0.0.fd00,0x50050763051bc74c,0x4010
401800000000 hvc_iucv=8 rd.zfcp=0.0.fc00,0x50050763050bc74c,0x4010401800000000 r
d.zfcp=0.0.fc00,0x500507630500c74c,0x4010401800000000 crashkernel=4096M LANG=en_
US.UTF-8 systemd.unit=rescue.target
...
Welcome to rescue mode! Type "systemctl default" or ^D to enter default mode.
Type "journalctl -xb" to view system logs. Type "systemctl reboot" to reboot.
Give root password for maintenance
(or type Control-D to continue):
```

In rescue.target mode, all of the file systems in /etc/fstab are mounted, but networking has not been started.

Load system targets on RHEL z/VM - FCP

To enter a different systemd target, from the IPL command just type the target you want in the systemd.unit= parameter. For example, to enter the emergency.target, use the following commands:

```
===> cp set loaddev portname 50050763 0500C74C lun 40104018 00000000 scpdata
systemd.unit=emergency.target
===>ipl fc00
```

Load system targets on RHEL z/VM - ECKD/EDEV

To load systemd targets from the IPL command when using DASD ECKD/FBA use the following commands from the 3270 terminal:

```
===> ipl 100 PARM systemd.unit=rescue
 00: zIPL v1.23.0-17.el7 interactive boot menu
 00:
 00:  0. default (linux)
 00:
 00:  1. linux
 00:
 00: Note:VM users please use '#cp vi vmsg <input>'
 00:
 00: Please choose (default will boot in 5 seconds):
 00: Booting default (linux)...
 00: Uncompressing Linux...
 ...
 Welcome to rescue mode! Type "systemctl default" or ^D to enter default mode.
 Type "journalctl -xb" to view system logs. Type "systemctl reboot" to reboot.
 Give root password for maintenance
 (or type Control-D to continue):
```

RESCUE mode

Create a copy of the **GENERIC PRM** file

```
===> copyfile generic prm a rescue prm a
```

Create a copy of the **REDHAT EXEC**

```
===> copyfile redhat exec d rescue exec a
```

Edit the **RESCUE EXEC** file, replacing the **GENERIC PRM** for the **RESCUE PRM** parameter:

```
/* */
'CL RDR'
'PURGE RDR ALL'
'SPOOL PUNCH * RDR'
'PUNCH KERNEL  IMG * (NOH'
'PUNCH RESCUE PRM * (NOH'
'PUNCH INITRD  IMG * (NOH'
'CH RDR ALL KEEP NOHOLD'
'I 00C'
```

Enter RESCUE mode

Edit the **GENERIC PRM** file

```
===> x rescue prm a
ro ramdisk_size=40000 cio_ignore=all,!condev
   ip=9.12.7.99::9.12.4.1:20:vmlnx2-4.itso.ibm.com:enccw0.0.0600:none
   rd.znet=qeth,0.0.0600,0.0.0601,0.0.0602,layer2=1
   nameserver=9.12.6.7 nameserver=9.12.6.6
   rd.zfcp=0.0.fc00,0x500507630500c74c,0x4010401800000000
   rd.zfcp=0.0.fd00,0x5005076305 10c74c,0x4010401800000000
   inst.repo=ftp://9.12.7.96/pub/rhel71
   inst.cmdline
rescue
```

Enter RESCUE mode

Run the **RESCUE EXEC** to start the rescue environment:

```
===> rescue exec a
00: 0000003 FILES PURGED
00: RDR FILE 0397 SENT FROM LINUX1   PUN WAS 0397 RECS 047K CPY  001 A NOHOLD NO
KEEP
00: RDR FILE 0401 SENT FROM LINUX1   PUN WAS 0401 RECS 0008 CPY  001 A NOHOLD NO
KEEP
00: RDR FILE 0405 SENT FROM LINUX1   PUN WAS 0405 RECS 329K CPY  001 A NOHOLD NO
KEEP
00: 0000003 FILES CHANGED
00: 0000003 FILES CHANGED
00: Uncompressing Linux...
00: Ok, booting the kernel.
00:
[    0.000000] Initializing cgroup subsys cpuset
[    0.000000] Initializing cgroup subsys cpu
[    0.000000] Initializing cgroup subsys cpuacct
es: 258048
[    0.000000] Kernel command line: ro ramdisk_size=40000 cio_ignore=all,!condev
                                  ip=9.12.7.99::9.12.4.1:20:vmlnx2-4.itso.ibm.
com:enccw0.0.0600:none            rd.znet=qeth,0.0.0600,0.0.0601,0.0.0602,laye
r2=1                              nameserver=9.12.6.7 nameserver=9.12.6.6
                                  rd.zfcp=0.0.fc00,0x500507630500c74c,0x401040
1800000000                        rd.zfcp=0.0.fd00,0x500507630510c74c,0x401040
1800000000                        inst.repo=ftp://9.12.7.96/pub/rhel71
                                  inst.cmdline
                                  rescue
...
```

Thank you
Danke
Grazie
Merci
謝謝
ありがとう
Obrigado!


plus.google.com/+RedHat


facebook.com/redhatinc


linkedin.com/company/red-hat


twitter.com/RedHatNews

youtube.com/user/RedHatVideos